Design and Error Analysis of a Vehicular AR System with Auto-Harmonization

Eric Foxlin, Thomas Calloway, and Hongsheng Zhang

Abstract—This paper describes the design, development and testing of an AR system that was developed for aerospace and ground vehicles to meet stringent accuracy and robustness requirements. The system uses an optical see-through HMD, and thus requires extremely low latency, high tracking accuracy and precision alignment and calibration of all subsystems in order to avoid mis-registration and "swim". The paper focuses on the optical/inertial hybrid tracking system and describes novel solutions to the challenges with the optics, algorithms, synchronization, and alignment with the vehicle and HMD systems. Tracker accuracy is presented with simulation results to predict the registration accuracy. A car test is used to create a through-the-eyepiece video demonstrating well-registered augmentations of the road and nearby structures while driving. Finally, a detailed covariance analysis of AR registration error is derived.

Index Terms—Inertial, augmented reality, calibration, registration, hybrid tracking, see through HMD, image processing, sensor fusion

1 Introduction

Recently there is an explosion of interest in Augmented Reality well beyond the research community. The popular press has adopted the vision of the pioneering researchers, in which AR will become an indispensable tool to augment human performance by providing enhanced situational awareness and visual guidance to complete tasks quickly and accurately without advance training.

For the past several years it seemed that the early focus on HMD-based AR had largely given way to tablet and phone AR, as the devices became widely available to consumers and advertisers saw the novelty of simple video AR as a way to reach them. However, with the advent of Google Glass and many other new see-through HMDs, there is a resurgence of interest in the original wearable AR paradigm, which in some sense can be considered the holy grail of AR because it leaves the user's hands free and can provide an always-on information display that is ready to provide augmentations quickly when they are needed.

With this renewed interest in HMDs comes a return to the thorny challenges that consumed researchers in the earlier years, mainly optical technologies to produce small comfortable HMDs with sufficient FOV, and head-tracking that can produce convincing spatio-temporal registration of augmentations to their corresponding physical objects in unprepared real-world environments (e.g., [3], [12], [18], [19], [21]). It is possible that AR will not achieve really widespread adoption until the registration appears "rock solid" with no noticeable "swim" [3]. There has been tremendous progress in vision-based tracking over the years. The ability to operate without markers has been demonstrated in many indoor and outdoor environments at impressive scale [12], [15], [18], and for video-see-through AR (such as tablets and phones) vision-based techniques also produce rock-solid registration with no noticeable swim or mis-registration. However optical see-through registration is a much harder problem because the view of the physical world cannot be delayed to match the view of the virtual augmentations, and the alignment cannot be simply matched up in a video image, which puts a much greater demand on absolute 6-DOF pose accuracy and relative calibration accuracy of the tracker to the display.

One of the main contributions of this paper is to describe the engineering of a system that achieves excellent spatio-temporal registration for a see-through HMD. The second main contribution is to describe the system architecture, motion-tracking algorithms, and harmonization techniques necessary to implement a precision AR system in a moving vehicle. The vast majority of AR tracking and computer vision papers focus on the grand challenge of markerless tracking for walking users in various indoor and outdoor environments. Indeed there are many important applications for users on foot and this problem absolutely needs to be solved. However, there are compelling applications for AR in various kinds of aircraft and ground vehicles as well. In fact, the very earliest head-tracked see-through HMDs were used on aircraft decades before the term "AR" was coined [11]. Vision-based tracking on vehicles presents very different challenges than for normal ground-based AR. The immediate visual surroundings are the interior of the vehicle, where use of markers may be perfectly practical. However, these markers (or any other visual features in the vehicle) are moving relative to the world frame where augmentations need to be stabilized, thus requiring careful engineering to track the vehicle, and the head relative to the vehicle, and combine it all with sufficient accuracy to achieve convincing optical see-through registration.

• *The authors are with the Thales Visionix, Inc.. E-mail: efoxlin@gmail.com, {thomas.calloway, hongsheng.zhang}@thalesvisionix.com.*

Manuscript received 18 Jan. 2015; revised 1 Aug. 2015; accepted 6 Sept. 2015. Date of publication 22 Sept. 2015; date of current version 26 Oct. 2015. Recommended for acceptance by S. Julier, R. W. Lindeman, and C. Sandor. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2015.2481385

The remainder of the paper is structured as follows. Section 2 presents the overall system architecture and design, including new algorithms that were developed to fuse inertial sensors that measure the head rotation relative to an inertial world-frame with optical corrections relative to a moving vehicle frame. Section 3 presents results from a very comprehensive model-based simulation of the hybrid tracking system, accounting for all errors from the tracking system and the harmonization process. Section 4 presents a validation experiment in an actual vehicle using a through-the-HMD video to demonstrate that the predicted registration results are achieved in practice. Section 5 is a statistical analysis of the registration error of the entire system including alignment errors between the HMD, head-tracker, fiducial constellation, and vehicle-mounted inertial navigation system (INS). Section 6 is the conclusion and discussion.

1.1 Previous Work

Many previous papers have addressed AR systems and tracking technologies for non-vehicular applications [12], [21]. Ferrin [6] surveys the variety of head-trackers that were in use on airplanes prior to 1991, which were mostly mechanical, optical and magnetic. Each type of head tracking technology has been demonstrated to feature its own set of strengths and weaknesses. Mechanical linkages are highly accurate but large and cumbersome [22]. Outside-in optical trackers can be compelling, but the installation of additional electronics (i.e., cameras) in a cockpit adds undesirable cost and complexity [8], [23]. Finally, magnetic systems are highly susceptible to electromagnetic interference. This makes their use particularly problematic in helicopters and other environments where the magnetic field might change. We chose to pursue hybrid optical-inertial technology in an attempt to minimize the aforementioned limitations knowing that aircraft rotation would add algorithmic complexity with the use of head worn inertial sensors.

We described a hybrid inertial helmet tracker for aircraft in [9], adding an inside-outside-in optical component to it in [8]. However the present system is descended much more closely from the InterSense IS-1200 VisTracker [7]. InterSense was acquired by Thales Visionix, and a significant effort has since been deployed to develop a much more robust version of the VisTracker called Hybid Optical-based Inertial Tracker (HObIT) for vehicular AR applications. A non-technical overview of the HObIT was provided in [1]. This is an extended version of our ISMAR paper [10], which describes the tracking algorithms in detail, including a new "auto-harmonization" approach for determining the precise alignment of the fiducial constellation to the vehicle inertial navigation system. It also describes a new fully integrated AR system that allows us to produce video of the augmentation of objects on or near the road in order to judge registration accuracy. This paper extends [10] by adding an analysis of the total system registration error budget, including the effects of mis-calibration in each subsystem. We derive the covariance matrix for the calibration errors in display frame as a function of covariances of both dynamic tracking errors and static but random calibration and alignment errors, accounting for the correlations that arise from a user-calibration process called boresighting. The approach can be adopted by others to quantify or predict registration error of AR systems that involve HMD calibration to align display optics and sensors.

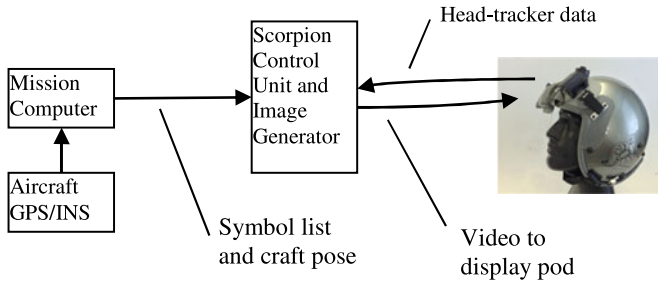

Fig. 1. Overall system architecture.

2 Design and Algorithms

2.1 System Architecture

The system used to produce the AR video in Section 5 is the "Scorpion" head-mounted display. As shown in Fig. 1, the Scorpion includes a ruggedized image generator and control unit which is mounted in the cockpit, as well as a display pod mounted on the helmet, plus the new HObIT head-tracker. The system was designed for use in military and civilian aircraft, and to improve the rate of adoption it was designed to be compatible with as many types of existing aircraft-installed equipment as possible. To make it compatible with different types of mission computer (MC) that may already be present in aircraft, the Scorpion display system implements only the generic head-tracking, rendering and display functions that are common to all vehicular AR systems, and none of the mission-specific functions such as targeting, cueing, enhanced vision or synthetic vision. The MC defines and downloads to the Scorpion image generator an arbitrary set of "symbols", which may include any 2D or 3D shapes involving line segments of any color or thickness and/or bitmaps. Each symbol may be specified by the MC to be ground-stabilized or head-stabilized or vehicle-stabilized. Once the symbols are downloaded, the Scorpion image generator renders them repeatedly at a 100 Hz HMD refresh rate, using new HObIT head-tracker data and new vehicle INS data for each frame.

The most important advantage of the new Hybrid Optical-based Inertial Tracker (HObIT) which is replacing a magnetic tracker in the Scorpion system is that it involves no cockpit-installed active devices (such as a magnetic source or optical cameras). Instead, all the inertial and optical sensors are in the self-contained HObIT sensor mounted on the helmet, which communicates to the cockpit-mounted Scorpion control unit through the same Helmet-Vehicle Interface (HVI) cable as the display pod. The only other parts are some peel-and-stick fiducial stickers that are placed on the canopy over the pilot or driver's head.

2.2 Optics and Image Processing

The biggest challenges for designing a version of the IS-1200 VisTracker for aircraft were in optics. We needed a very wide field fisheye lens because the camera is only about

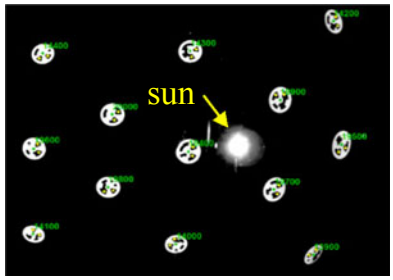

Fig. 2. Imaging fiducials with the sun directly behind them.

5-10 cm from the aircraft canopy. After extensive testing we found a lens that we were able to then optimize for good clear imaging at close range throughout a 140 degree FOV. Because the camera points straight up, the sun can be directly in the FOV, which makes for some very difficult dynamic range problems (see Fig. 2). Initially we tried to overcome this by using fiducials that get brighter proportionally when backlit by the sun. This worked very well in daylight, but in order to use them at night we added an illuminator and found we could not get sufficient brightness using wavelengths compatible with night vision goggles (NVGs).

We solved this problem by using retro-reflective stickers to overcome the inverse square loss from diffuse fiducials and return as much of the illumination to the lens as possible (Fig. 3). With retro-reflectors, we could illuminate the fiducials with NVG-compatible wavelengths and obtain high read rate throughout the wide FOV. This brought us back to the problem of fiducial readability in direct sunlight. Even with a tight bandpass filter installed behind the lens, the sun would bloom in the image whenever it was directly in view. We had to lower the exposure from our lowest usable setting of 1 millisecond to well below 50 microseconds in order to reject a satisfactory amount of sunlight. We were finally able to meet our design objectives of sunlight readability and NVG-compatibility with a unique illuminator design.

The image processing algorithms were developed and tested in the Mathworks Simulink development environment. The final "C" code was automatically generated and linked into our existing code running on a 1 GHz ARM processor located in the HObIT sensor itself. The tracker is able to capture and process entire 1,280 × 1,024 pixel images of fiducials at rates of up to 40 Hz. Fig. 4 below shows the high fiducial read rate of the HObIT tracker. By illuminating retro-reflective targets directly we avoid problems caused by ambient lighting such as shadows or sub-optimal exposures.

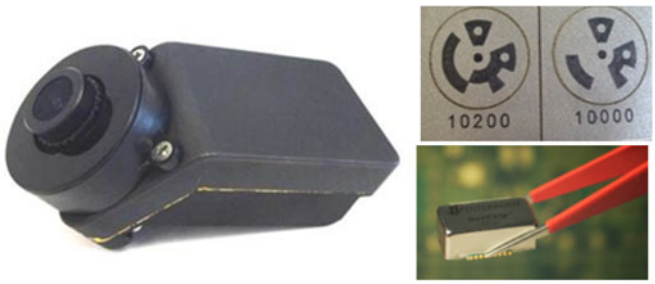

Fig. 3. HObIT tracker, retro-reflective fiducials and the InterSense NavChip inertial measurement unit that's inside.

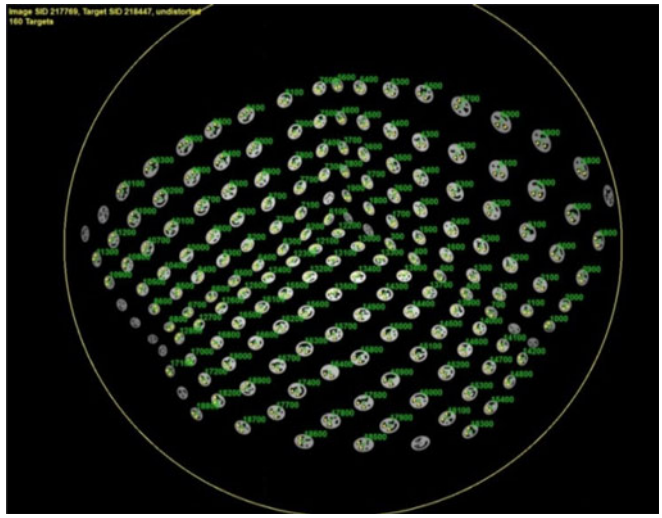

Fig. 4. Image captured from HObIT camera shows clean separation of retro-reflector fiducials from background and high fiducial read rate.

2.3 Camera Calibration, Fiducial Mapping and Vehicle Harmonization

2.3.1 *Camera Calibration*

The intrinsic distortion parameters of the HObIT's fisheye lens are calibrated by taking about 40 images of the three-walled fiducial-clad calibration fixture shown in Fig. 4. A modified version of the Caltech calibration toolbox has been adapted to fit an appropriate set of distortion parameters to model the lens with improved accuracy.

To solve the extrinsic calibration of the relative pose between the camera and the NavChip™ inertial measurement unit (IMU) inside the HObIT sensor module, an optical pose computed for each calibration frame is associated with a simultaneous pitch and roll measurement computed from the NavChip's sensors by an attitude and heading reference system (AHRS) algorithm. An optimization algorithm seeks the extrinsic parameters that cause all the gravity vectors from the AHRS to align with the downward direction of the calibration stand when transformed from the body axes of the IMU to the camera axes by the extrinsic rotation and then transformed to world frame by the camera pose rotation (Fig. 5).

2.3.2 *Fiducial Auto-Mapping*

A custom photogrammetric 3D mapping tool was developed to allow field technicians to accurately map a group of fiducials in a few minutes by simply scanning the HObIT sensor over the fiducial field. The algorithm bootstraps by using Nister's five-point algorithm [14] to create an initial reconstruction from five fiducials found in common between two views. It then alternates between triangulation to localize additional points, pose recovery to add additional camera poses, and bundle adjustment to refine the motion and structure parameters and discard outliers. The automapper automatically scales the map when the user enters the diameter of the largest fiducial in the map (it does support mixed fiducial sizes). After scaling, the reported point tightness values are typically around 30-70 microns for a 30 cm grid. To confirm that the self-reported precision values are realistic, we re-mapped the same grid multiple times and wrote a

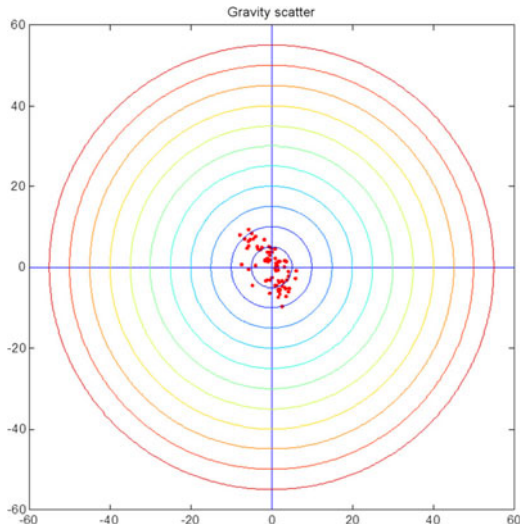

Fig. 5. Scatter plot of the world-frame gravity vectors after extrinsic parameter optimization. Each circle is 5 mrad.

point-cloud comparison tool to align the two clouds and quantify discrepancies. The 1-sigma 3-dimensional error (root-sum-square of the 1-sigma errors in each axis) was 43.55 microns for the 30 cm constellation shown in Fig. 6, corresponding to accuracy of about 1:7,000 of the extent of the constellation. We have obtained accuracies ranging from 1:5,000 to 1:20,000 for constellations of different sizes.

2.3.3 Vehicle Harmonization

The final and most difficult step of installation is harmonization. In general for avionics "harmonization" is the process of aligning the axes of various aircraft systems with one another, such as the inertial navigation system, the heads-up-display (HUD), the HMD tracking system reference, sensor pods or targeting pods, and weapons. We developed a variety of tools and methods to align the fiducial constellation with the aircraft axes, or more specifically with the platform INS axes since the INS is the reference frame from which symbol generators are driven. When the aircraft contains a HUD, we assume the HUD is already aligned with the p-frame of the INS and we use a specially developed tool containing a collimated optical scope with a HObIT sensor aligned on top. By pointing the scope at the watermark in the HUD it can be aligned with the platform x-axis, and at the same time the HObIT looks up at the fiducials and

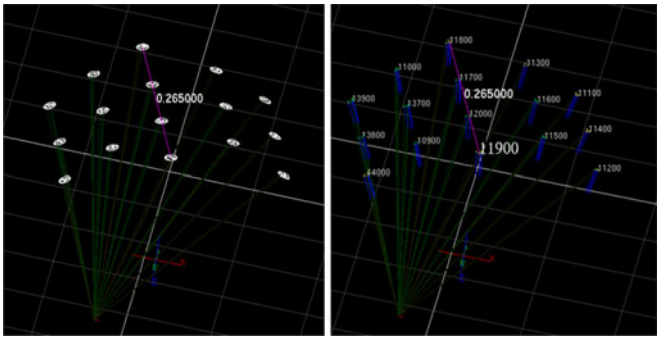

Fig. 6. Screenshots from the auto-mapping tool. Right image shows how normal vectors are automatically computed by finding the locations of the three "eyes" in each fiducial and calculating the cross product.

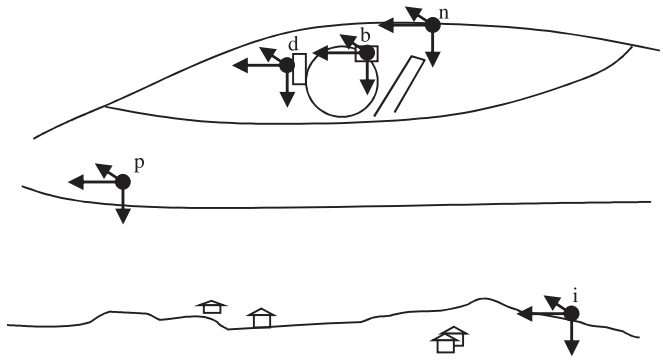

Fig. 7. Illustration of a fighter jet cockpit with coordinate axes associated with the platform INS (p), the Scorpion display (d), the sensor body (b), the fiducial constellation (n) and the ground (i).

determines the pose of the scope relative to n-frame, from which we solve the rotation of the n-frame w.r.t. the p-frame.

Unfortunately many aircraft and most ground vehicles lack a HUD. In those cases we had to resort to physically measuring and aligning to multiple reference points on the aircraft. This process is very cumbersome and time-consuming because it involves the use of external sighting equipment and alignment processes that vary from platform to platform. Additionally, observable reference points on an aircraft are rarely well aligned to the aircraft's internal navigation system. This makes the manual alignment process almost impossible to perform with sufficient accuracy. The following sections will outline a solution we developed which eliminates the process and greatly improves accuracy.

2.4 Tracking, Auto-Harmonization and Delay Estimation Algorithms

Fig. 7 above illustrates the use of the Scorpion system in a cockpit as well as the five associated coordinate frames that are shown in Table 1. Vectors are denoted by lowercase

TABLE 1
Five Coordinate Systems

i-frame	The i-frame is an inertial reference frame, which for our purposes is a local-level North-East-Down (NED) frame on the ground below the aircraft that rotates sufficiently slowly to be considered an inertial frame.
p-frame	The aircraft "platform INS" frame. The "platform INS" is the inertial navigation system that supplies pose data to the mission computer and in turn to the Scorpion display system.
n-frame	The reference frame of the tracking system. For a magnetic tracker the n-frame has its origin in and axes nominally aligned with the source coil assembly. For HObIT, the n-frame has its origin at one of the fiducials and its axes are roughly aligned to the aircraft axes during ground harmonization procedures.
b-frame	The body frame of the tracker sensor. In the case of HObIT, the b-frame is defined by the NavChip inside the sensor assembly, which is mounted upside-down, backwards and tilted relative to the helmet.
d-frame	Display frame defined by the lightguide optical element (LOE) or "paddle" on the Scorpion display pod.

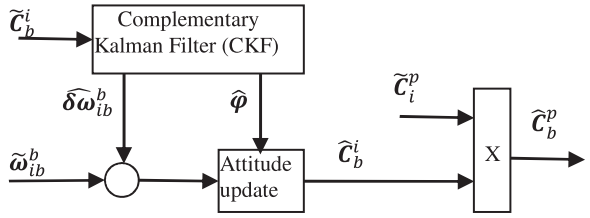

Fig. 8. Algorithm for tracking relative to i-frame, then converting output to p-frame for legacy tracker compatibility.

bold letters and matrices by uppercase bold. If a vector is expressed in a particular coordinate system it is denoted with a superscript designating one of the above five frames. A rotation matrix has a subscript and a superscript and transforms a vector from the subscript frame to the superscript frame. A hat above a quantity refers to an estimated value, a tilde above means a measured value, and the unadorned symbol represents the true value.

2.4.1 Basic Orientation Tracking Filter

The Extended Kalman Filter (EKF) for the HObIT is greatly simplified compared to the VisTracker. Because the VisTracker had a rolling-shutter image sensor, it had to process each individual fiducial measurement separately at a different point in time, using a highly nonlinear bearings-only measurement model which was a function of position as well as orientation [7]. Due to the global shutter imager and the much faster processing element (ARM Cortex A8 at 1 GHz), the HObIT is able to simultaneously capture and decode up to 20 fiducials at frame rate. For every frame, it solves for pose using a modified version of the OpenCV pose recovery algorithm, which results in a direct measurement of the rotation that can be used to correct gyro drift. Therefore, the head orientation can be tracked independently from position using just gyros and camera pose measurements and a very simple 6-state Complementary Kalman Filter (CKF) to estimate the rotation errors and gyro biases as shown in Fig. 8.

The head-tracker gyros measure $\tilde{\omega}_{\mathrm{ib}}^{\mathrm{b}}$, the angular rates of the sensor (b-frame) relative to the ground (i-frame). Rather than use the more complicated differential inertial algorithm described in [9], we elected to simply integrate the angular rates to estimate head orientation with reference to the ground, $\hat{\mathrm{C}}_{\mathrm{b}}^{\mathrm{i}}$, and use the complementary filter to directly correct the errors of this ground-referenced orientation as well as the gyro biases. As a final step, we pre-multiply the result by the inverse of the aircraft attitude supplied from the platform INS in order to provide the head-tracker orientation w.r.t. the aircraft platform axes (which is required by legacy mission computers that were designed for non-inertial trackers).

Because the optical subsystem measures the pose of the sensor b-frame with respect to the fiducial constellation n-frame, $\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{n}}$, we need to feed the CKF a pseudo-measurement relative to the ground formed as follows:

$$\tilde{\boldsymbol{C}}_b^i = \boldsymbol{C}_p^i \boldsymbol{C}_n^p \tilde{\boldsymbol{C}}_b^n. \tag{1}$$

Here $\mathrm{C}_{\mathrm{p}}^{\mathrm{i}}$ is the attitude of the vehicle returned from the platform INS, $\tilde{\mathrm{C}}_{\mathrm{b}}^{\mathrm{n}}$ is the measurement of the HObIT sensor pose relative to the fiducial constellation frame, and $\mathrm{C}_{\mathrm{n}}^{\mathrm{p}}$ is the alignment rotation relating the constellation reference frame to the platform INS frame, which was determined during the harmonization process described above.

The CKF is simply a linear Kalman Filter that estimates the small errors that accumulate in the attitude update algorithm, together with the gyro biases that are partly responsible for creating those errors. The state vector is $\delta x = \begin{bmatrix} \boldsymbol{\varphi}^T & \boldsymbol{\delta\omega}^T \end{bmatrix}^T$ where $\boldsymbol{\delta\omega}$ are the gyro bias errors, and the small-angle rotation error vector $\boldsymbol{\varphi}$ is defined by

$$\hat{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}} = (\boldsymbol{I} - [\boldsymbol{\varphi}\times])\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{i}}. \tag{2}$$

The notation $[\boldsymbol{\varphi}\times]$ represents the skew-symmetric matrix or cross product operator:

$$[\boldsymbol{\varphi}\times] = S(\boldsymbol{\varphi}) = \begin{bmatrix} 0 & -\varphi_z & \varphi_y \\ \varphi_z & 0 & -\varphi_x \\ -\varphi_y & \varphi_x & 0 \end{bmatrix}. \tag{3}$$

From [16], the time update equation for the CKF is

$$\begin{bmatrix} \boldsymbol{\varphi} \\ \boldsymbol{\delta\omega} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{I} & -\Delta t \boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{i}} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi} \\ \boldsymbol{\delta\omega} \end{bmatrix}_k + \boldsymbol{w}_k. \tag{4}$$

To simplify the derivation of the measurement update equation, we define the measurement for the CKF not as the quantity $\tilde{\mathrm{C}}_{\mathrm{b}}^{\mathrm{i}}$ calculated in (1), but as a residual calculated from it as follows:

$$\boldsymbol{z} \triangleq S^{-1}(\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}} \hat{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}^T} - \boldsymbol{I}). \tag{5}$$

We have introduced an inverse skew-symmetric matrix operator that extracts the corresponding vector elements from a skew-symmetric matrix. We know that the input matrix will be very close to skew symmetric because the measurement $\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}}$ is derived from a very accurate optical pose recovery combined with a very accurate (sub-milliradian) aircraft attitude from an INS, and a harmonization matrix that was hopefully determined accurately in advance. Ignoring the small measurement error, we can expand (5) to get

$$\begin{aligned} \boldsymbol{z} &\triangleq S^{-1}\left(\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}} \hat{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{i}^T} - \boldsymbol{I}\right) \\ &\approx S^{-1}\left(\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{i}} \boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{b}} (\boldsymbol{I} + [\boldsymbol{\varphi}\times]) - \boldsymbol{I}\right) \\ &= S^{-1}([\boldsymbol{\varphi}\times]) \\ &= \boldsymbol{\varphi}, \end{aligned}$$

which simplifies the CKF measurement update equation to

$$\begin{aligned} \boldsymbol{z}_k &= \boldsymbol{H}\,\boldsymbol{\delta x}_k + \boldsymbol{v}_k \\ \boldsymbol{H} &= [\,\boldsymbol{I} \quad 0\,]. \end{aligned} \tag{6}$$

The $\boldsymbol{w}_k$ in (4) and $\boldsymbol{v}_k$ in (6) represent white process noise and measurement noise vectors, respectively.

2.4.2 Filter Augmentation for Auto-Harmonization

We initially implemented the CKF as described in the preceding section and were able to obtain reasonably good results, but we found in the first set of flight tests that the overall system error budget was dominated by errors in the harmonization alignment matrix $\mathrm{C}_{\mathrm{n}}^{\mathrm{p}}$ which had been determined during installation using the techniques described in

Section 2.3. We worked hard to improve our manual harmonization accuracy with various techniques and procedures, but achieving high accuracy in ground harmonization requires delicate and expensive precision surveying equipment which would be logistically difficult to maintain at all the necessary locations to service a fleet, and lengthy elaborate manual operations requiring very patient and skilled technicians. Worse yet, manual measurement-based harmonization procedures can only facilitate aligning the fiducial constellation with visible features on the airplane (such as a crosshair in a HUD or a boresight reticle unit (BRU)). The accuracy of the AR system however relies on the alignment of the fiducials with the platform INS which is typically installed deep in the avionics bay and cannot be directly measured. This means our manual harmonization relies on the accuracy of some previous harmonization that should have aligned the INS with the HUD. Unfortunately in many cases the vehicle had no HUD with which to align our constellation.

With that motivation we set out to augment the Kalman filter in the previous section with three additional states to estimate the harmonization error angles $\boldsymbol{\psi}$, which are defined analogously to the tracker error angles in (2):

$$\hat{\boldsymbol{C}}_{\boldsymbol{n}}^{\boldsymbol{p}} = (\boldsymbol{I} - [\boldsymbol{\psi}\times])\boldsymbol{C}_{\boldsymbol{n}}^{\boldsymbol{p}}. \tag{7}$$

These states are constant, so the time propagation equation remains basically the same with the added states:

$$\begin{bmatrix} \boldsymbol{\varphi} \\ \boldsymbol{\delta\omega} \\ \boldsymbol{\psi} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{I} & -\Delta t\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{i}} & 0 \\ 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi} \\ \boldsymbol{\delta\omega} \\ \boldsymbol{\psi} \end{bmatrix}_{k} + \boldsymbol{w}_k. \tag{8}$$

To get the new measurement equation, substitute the right hand side of (7) for $\boldsymbol{C}_n^p$ in (1), then substitute that into (5) to get

$$\begin{aligned} z &\triangleq S^{-1}\left(\tilde{C}_b^i \hat{C}_{\boldsymbol{b}}^{\boldsymbol{i}^T} - \boldsymbol{I}\right) \\ &= \boldsymbol{S}^{-1}\left(\boldsymbol{C}_{\boldsymbol{p}}^{\boldsymbol{i}} \hat{\boldsymbol{C}}_{\boldsymbol{n}}^{\boldsymbol{p}} \tilde{C}_b^n \hat{C}_{\boldsymbol{b}}^{\boldsymbol{i}^{\boldsymbol{T}}} - \boldsymbol{I}\right) \\ &\approx S^{-1}\left(\boldsymbol{C}_p^i(\boldsymbol{I} - [\widehat{\boldsymbol{\psi}}\times])\boldsymbol{C}_n^p \tilde{C}_b^n \boldsymbol{C}_b^i(\boldsymbol{I} + [\boldsymbol{\varphi}\times]) - \boldsymbol{I}\right) \\ &= S^{-1}\left(-\boldsymbol{C}_p^i[\boldsymbol{\psi}\times]\boldsymbol{C}_i^p + [\boldsymbol{\varphi}\times]\right) \\ &= -\boldsymbol{C}_p^i\boldsymbol{\psi} + \boldsymbol{\varphi}. \end{aligned} \tag{9}$$

The last line takes advantage of the fact that the similarity transformation to the i-frame of the p-frame cross product operator $[\boldsymbol{\psi}\times]$ must be geometrically equivalent to an i-frame cross product operator using $\boldsymbol{\psi^i} = \boldsymbol{C}_p^i\boldsymbol{\psi}$. Once again we end up with an elegantly simple CKF measurement update equation that relates the measurement residual z to both the tracker error $\boldsymbol{\varphi}$ and an effectively static measurement bias $\boldsymbol{\psi}$:

$$z_k = \boldsymbol{H}\,\delta x_k + \boldsymbol{v}_k \tag{10}$$

with

$$H = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{C}_p^i \end{bmatrix} \tag{11}$$

.

Because $\boldsymbol{C}_p^i$ varies with the vehicle attitude, the observability Gramian becomes full rank after the vehicle takes off and begins to maneuver, so only then will both the head orientation and harmonization error covariances converge to small values.

2.4.3 Filter Augmentation for Compensating Delayed INS Data

We implemented the auto-harmonization feature in the previous section and verified it in simulation and car tests. Unfortunately the first time we went to flight test it in a helicopter, we encountered an unexpected problem. The flight test data collection computer was capturing the platform INS data at 25 Hz (which is an acceptable rate because the HObIT tracking algorithms use head-mounted gyros to directly track pilot head orientation relative to the ground at 200 Hz without using any platform INS data in the primary AR data path). However the data was being delayed by an unknown amount that the pilot estimated as upwards of 200 ms. With that much delay, the measurement updates in the Kalman filter would receive measurement errors significantly larger than the tuned measurement noise matrix resulting in suboptimal performance. Rather than retune the filter to expect vastly more measurement noise, we determined that given the slow dynamics of an aircraft it would be possible to remove most of the error by forward predicting the aircraft attitude if we only knew the amount of latency we needed to compensate. We took the approach of trying to automatically estimate and adapt to the data latency by adding one additional state to the CKF to estimate the error of the prediction interval currently in effect for compensating the measurement data latency.

First, we added a block to perform forward prediction of the platform INS attitude $\boldsymbol{C}_p^i$ by a rotation angle of $\boldsymbol{\omega}_{ip}^p\Delta t$ where platform angular velocity $\boldsymbol{\omega}_{ip}^p$ is estimated from the last two samples of platform attitude, and Δt is initialized to 50 ms with an uncertainty of 100 ms. Finally we replace $\boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{p}}$ in the measurement equation with

$$\hat{\boldsymbol{C}}_{\boldsymbol{i}}^{\boldsymbol{p}} = \left(\boldsymbol{I} - \boldsymbol{\tau}\left[\boldsymbol{\omega}_{ip}^p\times\right]\right)\boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{p}}, \tag{12}$$

which results in our final 10-state measurement model

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{C}_p^i & \boldsymbol{C}_p^i\boldsymbol{\omega}_{ip}^p \end{bmatrix}. \tag{13}$$

Although we have not seen this simple approach to delay estimation and compensation in the literature on Kalman filtering with delayed measurements, it works very well as shall be seen below.

3 Simulation Accuracy Analysis

3.1 Simulation

The main algorithmic engine for HObIT, called sfCore, was developed using model-based design in Simulink. Therefore, much effort was spent up front developing a high-fidelity simulation harness so the algorithms could be continuously tested with realistic data. Fig. 9 shows the architecture of the simulation environment. The sfCore algorithm is run as a software-in-the-loop (SIL) block of compiled C code that was auto-generated from a Simulink model and runs within the simulation model. A key aspect

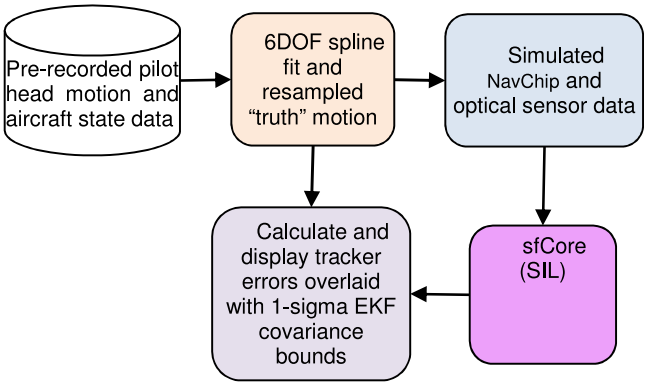

Fig. 9. Outline of the simulation framework.

of the simulation is the use of splines to generate high sample rate (five KSPS) truth data from the 100 SPS recorded head motion and aircraft motion that was obtained on previous test flights. The high sample rates are needed in order to simulate in detail the internal oversampling and integration algorithms of the NavChip IMU in order to produce simulated inertial measurements with the same drift characteristics as the actual HObIT. Careful verification was performed by generating simulated perfect IMU measurements ($\Delta\theta$ and ΔV) with no noise or error sources, and showing that 6DOF strapdown INS integration algorithms applied to these measurements reconstruct the truth trajectory perfectly.

The simulation has been run frequently over two years with many different motion datasets captured from different phases of flight on different types of aircraft. At every stage we use the simulation results to tune the sfCore algorithms to achieve consistent performance. Consistency is monitored in two ways: the monte carlo simulated errors should be within the 1-sigma covariance bounds about 66 percent of the time, and the normalized estimation error squared (NEES) exceeds its 1 percent chi-square value less than 1 percent of the time.

For brevity we here show results from just one simulation run, which consists of 5 minutes of data starting before taxi and take-off on a fast jet. For this run the auto-harmonization filter is turned on, with initial harmonization angle uncertainties of 10 degrees in roll and five degrees each in pitch and yaw. (This is enough to allow the installer to simply

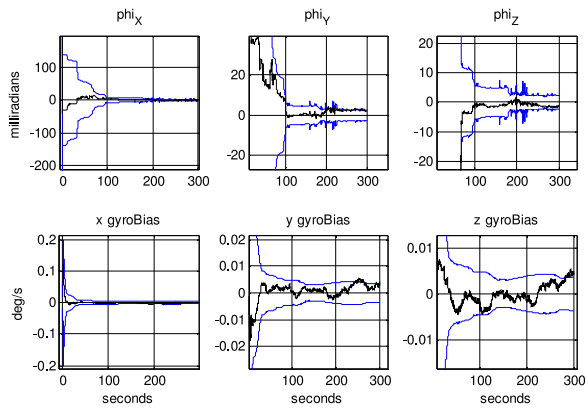

Fig. 10. Simulated angular and gyro bias errors and covariances for a five minute take-off sequence.

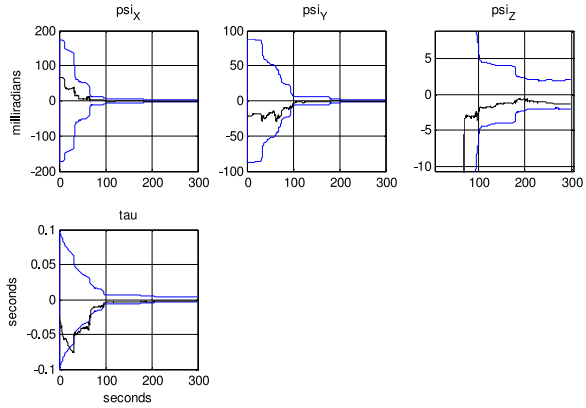

Fig. 11. Simulated harmonization angle and time delay errors and covariances for a 5 minute take-off sequence.

line up the constellation with the aircraft axes by eye.) Figs. 10 and 11 show plots of the errors and their one-sigma covariance bounds for all ten states of the Kalman filter. The first row of Fig. 10 shows all three components of $\boldsymbol{\varphi}$, which are the errors in the head-tracker's estimation of $\boldsymbol{C}_b^i$. Since $\boldsymbol{C}_b^i$ represents head-tracker orientation with respect to the ground not the constellation, these errors actually include the combined effects of tracker error and harmonization error, as the measurements that are correcting $\boldsymbol{C}_b^i$ contain the harmonization error in them. That is why they start off large until the plane takes off and the auto-harmonization states (shown on the first row of Fig. 11) converge. The plots of the y and z axis errors are zoomed in to show that after a few maneuvers lasting only a couple minutes from take-off, the harmonization errors and the tracking errors converge down to well below 5 mrad. The second row of Fig. 11 shows the convergence of the delay estimation state, which successfully estimates the simulated random delay to within two milliseconds.

The second row of Fig. 10 shows that the gyro bias estimates converge very quickly (due to the extremely low noise gyros in the NavChip) and hover around 0.005 deg/s rms. This is sufficiently low to achieve the desired drift rate when out of optical lock of less than 10 mrad/minute, as confirmed by real experimental data in Fig. 12. Note that drift performance while out of optical lock (meaning the camera cannot see at least four fiducials) is actually quite important because in some aircraft it is desirable to use as few fiducials as possible, which results in the pilot frequently moving the helmet to a position where the HObIT can't see enough fiducials. In helicopters they even want to use the tracker while leaning out the window!

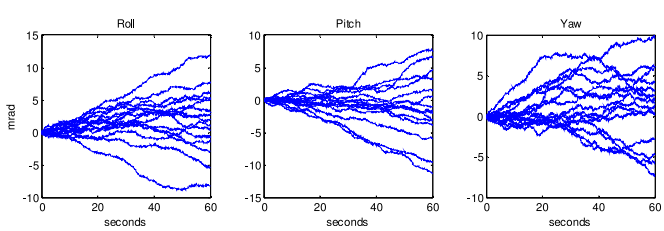

Fig. 12. One minute random gyro drift trials while out of optical lock (real data, not simulation).

Fig. 13. Fiducials installed on sun roof over passenger seat.

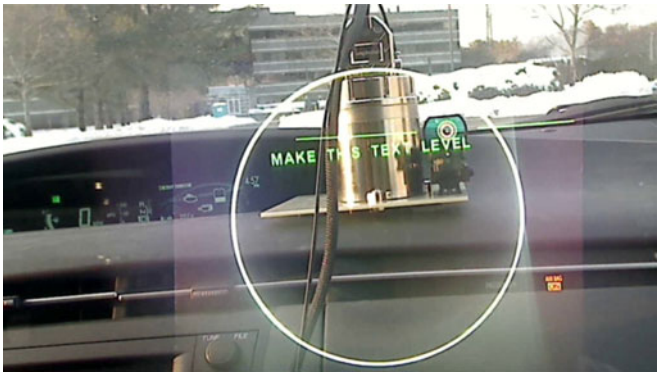

Fig. 14. GPS/INS installed on dash together with a small rifle scope for boresighting.

Fig. 15. Webcam mounted behind Scorpion display paddle to record the video.

4 Experimental Verification in a Car

To verify that the actual performance of the system matches that which is seen in simulation, we have developed a test system that we deploy in a Toyota Prius. The part that is installed in the car consists of a field of fiducials on the sunroof above the passenger seat (Fig. 13) and a Systron-Donner SDN500 GPS-aided INS that we placed on the dashboard to play the role of the platform INS (Fig. 14).

To capture the accompanying video footage, we mounted a webcam behind the Scorpion display paddle as shown in Fig. 15. Thus the video shows exactly what the eye sees when wearing the display and provides a good way to judge the registration capabilities of the system. We suggest you watch it, as the registration is much more impressive in dynamic motion due to the low latency and accurate prediction capabilities of the inertial tracker. However Fig. 16 shows one still frame from the video to give a sense of the content.

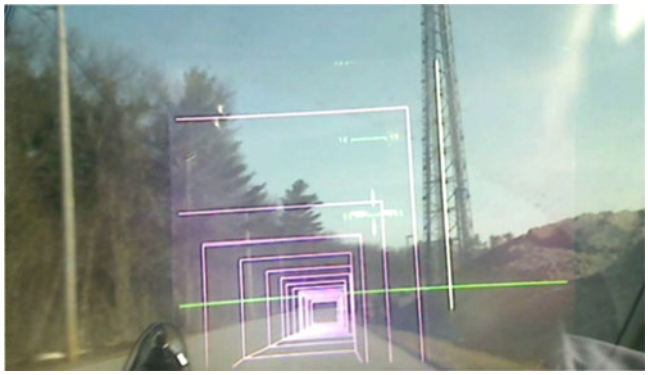

Fig. 16. One frame from the video showing how we augment a radio tower and place a "highway-in-the-sky" type of display on the road.

TABLE 2
Three Auto-Harmonization Trials Produce Consistent Results

	psiX (deg)	psiY (deg)	psiZ (deg)	Tau (ms)
Trial 1:	1.87	2.91	0.11	20.2
Trial 2:	1.89	2.89	0.48	21.2
Trial 3:	1.88	2.86	0.37	21.9

To draw the ground-fixed symbols such as the tower and road augmentations, the Scorpion control unit combines the car position and orientation measured by the platform INS with the head-tracker data to determine the head pose in geographic coordinates. Because the car position is determined by GPS, augmentations close to the car can sometimes appear to jump around by up to a meter, but objects at a distance look pretty good all the time.

Fig. 14 illustrates how we "boresight" the system to determine the rotation matrix between the HObIT sensor and the Scorpion display paddle (which is mounted on a hinge to allow the pilot to adjust for most comfortable positioning of the eyebox). A small collimated rifle sight is mounted with and aligned to the INS to take the place of a HUD. After adjusting the paddle mounting on the helmet, the user adjusts the three boresight angle offsets to nudge the green circle back to alignment with boresight mark and level the virtual horizon with the real horizon (or alternatively make a virtual vertical match a real one).

In addition to making a video to show off the spatio-temporal registration of the system, our road test included evaluation of the performance of the auto-harmonization and auto-time-delay-compensation algorithms whose simulated results are shown in Fig. 11. We started the system from scratch (wide open initial covariances) on three different drives and checked the consistency of the final harmonization angle and time delay estimates. The results are in Table 2 above. As can be seen, there was excellent agreement, well within the covariance bounds of the estimator after a few minutes of driving. This does not prove the results are accurate, but there seems to be no reason why they would consistently converge to the same value if the system was not working in close accord to the simulation.

5 Registration Error Analysis

The simulation described in Section 3 provides the covariance of the error states $\boldsymbol{\varphi}$, which are the errors in the head-tracker's estimation of C_b^i, including combined effects of tracker error and auto-harmonization error for a particular run. Unfortunately it is not straightforward from this to understand the AR registration errors that the pilot will see when looking in any particular direction, which involves a complicated interplay between the various error sources making up the tracker error budget and the boresighting process which serves to cancel out the repeatable parts of the tracking error in the frontal direction. Furthermore, pointing error is a two-dimensional error and should be characterized with a distance-RMS (DRMS) metric of the line-of-sight (LOS) vector which is computed from the tracker orientation output and the boresight transformation. The first goal of the analysis in this section is to determine the DRMS pointing accuracy for various helmet azimuth and elevation look directions. A covariance analysis is used to provide a statistical characterization of the system accuracy including both time-varying tracking errors that occur throughout the flight, as well as tracker-to-tracker, install-to-install, and flight-to-flight variations that occur due to one-time errors in calibration, harmonization, boresighting, etc. This analysis is needed to determine the statistical pointing accuracy of the system because it is not possible to collect and analyze enough different flights with different trackers and different installation parameters to determine the accuracy empirically, even if there were a way to measure the aiming accuracy of the system during flight.

5.1 Symbol Generation Pipeline

Let $\boldsymbol{v^i}$ represent the line-of-sight unit vector from the aircraft to a distant target with a symbol (augmentation) overlaid on it, expressed in i-frame. This gets converted to p-frame using $\boldsymbol{v^p} = C_i^p \boldsymbol{v^i}$, where C_i^p is the transpose of the rotation matrix C_p^i that it receives from the aircraft INS.

The Scorpion system then converts the symbol vector from p-frame to d-frame using:

$$\boldsymbol{v^d} = C_{\boldsymbol{b}}^{\boldsymbol{d}} C_{\boldsymbol{n}}^{\boldsymbol{b}} C_{\boldsymbol{p}}^{\boldsymbol{n}} \boldsymbol{v^p}. \tag{14}$$

This vector is then projected onto the 2D display with a perspective projection focal length chosen to scale the image to match the FOV of the display. Even if the FOV scaling is not quite perfect, this only affects registration in the periphery of the display, and the rest of this analysis will only consider registration accuracy at the center of the display, which is not affected by the FOV parameter.

C_n^p is the harmonization rotation. For HObIT, the n-frame is aligned at least roughly parallel to the p-frame during installation/mapping procedures, so the C_n^p that gets estimated by auto-harmonization should be within a few degrees of the identity matrix. The harmonization error angles vector ψ is defined by the small-angle rotation matrix approximation in Equation (7).

C_b^n is the orientation output of the tracker sensor with respect to its fiducial constellation, which varies dynamically with head motion. The tracking error ϕ, which may be due to any combination of optical and inertial calibration errors, noise, drift and wander, is defined by:

$$\hat{C}_b^n = (I - [\boldsymbol{\phi} \times]) C_b^n. \tag{15}$$

Note that in this section we are using ϕ for the error in C_b^n, unlike Equation (2) where it was used for the error in $C_{\boldsymbol{b}}^{\boldsymbol{i}}$.

C_b^d is the boresight rotation matrix which defines the orientation of the tracking sensor, b, relative to the display paddle, d. The boresight error angles vector β is defined by

$$\hat{C}_b^d = (I - [\boldsymbol{\beta} \times]) C_b^d. \tag{16}$$

Substituting these definitions into Equation (14) and transposing them as necessary, the calculated position to display a symbol on the target at $\boldsymbol{v}^p$ is:

$$\begin{aligned} \hat{\boldsymbol{v}}^d &= \hat{C}_b^d \hat{C}_n^b \hat{C}_p^n \hat{\boldsymbol{v}}^p \\ &= (I - [\boldsymbol{\beta} \times]) C_b^d C_n^b (I + [\boldsymbol{\phi} \times]) C_p^n (I + [\boldsymbol{\Psi} \times]) \hat{\boldsymbol{v}}^p. \end{aligned} \tag{17}$$

The symbol position error (in 3D), discarding terms that are second order in small error quantities, is:

$$\begin{aligned} \tilde{\boldsymbol{v}}^d &= \hat{\boldsymbol{v}}^d - \boldsymbol{v}^d \\ &= (I - [\boldsymbol{\beta} \times]) C_b^d C_n^b (I + [\boldsymbol{\phi} \times]) C_p^n (I + [\boldsymbol{\Psi} \times]) \hat{\boldsymbol{v}}^p - C_b^d C_n^b C_p^n \boldsymbol{v}^p \\ &\approx -[\boldsymbol{\beta} \times] C_p^d \boldsymbol{v}^p + C_n^d [\boldsymbol{\phi} \times] C_p^n \boldsymbol{v}^p + C_p^d [\boldsymbol{\Psi} \times] \boldsymbol{v}^p + C_p^d \boldsymbol{\delta v^p} \\ &\approx -[\boldsymbol{\beta} \times] C_p^d \boldsymbol{v}^p + C_n^d C_p^n [\boldsymbol{\phi} \times] \boldsymbol{v}^p + C_p^d [\boldsymbol{\Psi} \times] \boldsymbol{v}^p + C_p^d \boldsymbol{\delta v^p} \\ &= -\boldsymbol{\beta} \times \boldsymbol{v}^d + \left(\boldsymbol{\phi^d} + \boldsymbol{\Psi^d}\right) \times \boldsymbol{v}^d + \boldsymbol{\delta v^d}, \end{aligned} \tag{18}$$

where the third line introduces $\boldsymbol{\delta v^p} \equiv \hat{\boldsymbol{v}}^p - \boldsymbol{v}^p$, and the fourth line takes advantage of the approximate commutativity of small angle rotations assuming that C_p^n is close to identity (which is always the case for HObIT installs). The fifth line is just a change of basis from p-frame to d-frame for the second and third terms.

5.2 Solving for β Through Pilot Boresight

The three error terms are not independent because the pilot boresight procedure adjusts β to make the visible error exactly zero at the location of the boresighting target, which may be either the watermark in a Head-Up Display (HUD), or the crosshair in a Boresight Reticle Unit. Let v_1^p be the true location of the boresighting reference mark and $\hat{v}_1^p$ be the nominal position determined by design or measurement, which is used in the software to generate the boresight symbol in the Scorpion. They are not always exactly the same. The alignment accuracy at the center of a commercial HUD is typically +/-3 mrad, while refractive HUDs with integrated combiners (i.e., F-16) are capable of achieving +/–1.5 mrad [20]. However, BRUs are not aligned as accurately as HUDs.

When the pilot does the boresight adjustments, s/he is effectively driving $\tilde{v}^d$ to zero in Equation (18), which gives us the constraint:

$$\left[\boldsymbol{v}^d \times\right] \boldsymbol{\beta} = \left[\boldsymbol{v}^d \times\right] \left(\boldsymbol{\phi^d} + \boldsymbol{\Psi^d}\right) - \boldsymbol{\delta v^d},$$

which we shall write as :

$$[v^d\times]\boldsymbol{\gamma}^d = \boldsymbol{\delta v^d}, \tag{19}$$

where

$$\boldsymbol{\gamma^d} \equiv \boldsymbol{\phi^d} + \Psi^{\boldsymbol{d}} - \boldsymbol{\beta}. \tag{20}$$

We wish to solve this for $\boldsymbol{\gamma^d}$ from which we can get $\boldsymbol{\beta}$, but the rank of $[v^d\times]$ is only 2, so there is not a unique solution. This is obvious considering that by looking at just one point the pilot cannot resolve roll, so to fully resolve the boresight transformation the pilot also performs a roll adjustment which requires displaying a line segment or at least one additional point displaced a bit from the watermark.

Mathematically we can model the pilot's boresight adjustments as minimizing the error at two points, v_1 at the center of the watermark (in a HUD) or the crosshair (in a BRU), and v_2 displaced a small distance ε to the right of v_1. For example, this could be the end of the right extension of the mark. To minimize the differences at two points, we stack two constraints of the form of Equation (4), which results in a linear equation having full rank that can be solved by least squares:

$$\begin{bmatrix}[\boldsymbol{v_1^d}\times]\\ [\boldsymbol{v_2^d}\times]\end{bmatrix}\boldsymbol{\gamma^d} = \begin{bmatrix}\boldsymbol{\delta v^d}\\ \boldsymbol{\delta v^d}\end{bmatrix}, \tag{21}$$

$$\boldsymbol{\gamma^d} = \left(\begin{bmatrix}[\boldsymbol{v_1^d}\times]\\ [\boldsymbol{v_2^d}\times]\end{bmatrix}^T\begin{bmatrix}[\boldsymbol{v_1^d}\times]\\ [\boldsymbol{v_2^d}\times]\end{bmatrix}\right)^{-1}\begin{bmatrix}[\boldsymbol{v_1^d}\times]\\ [\boldsymbol{v_2^d}\times]\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\delta v^d}\\ \boldsymbol{\delta v^d}\end{bmatrix}, \tag{22}$$

$$\begin{aligned}\boldsymbol{\beta} &= \boldsymbol{\phi^d} + \Psi^{\boldsymbol{d}} - \boldsymbol{\gamma^d}\\ &= C_p^d(\boldsymbol{\phi} + \Psi - \boldsymbol{\gamma}).\end{aligned} \tag{23}$$

Note that $\boldsymbol{\gamma}$ without a superscript is in the p-frame and can be thought of as the error rotation vector of the HUD or BRU misalignment with respect to the p-frame.

5.3 Line-of-Sight Error

Now, given tracker and harmonization errors ϕ and $\boldsymbol{\psi}$, and HUD/BRU alignment error $\boldsymbol{\gamma}$, we can use (23) to find the boresight errors $\boldsymbol{\beta}$ that the pilot will dial in to cancel them, and then we can plug all these errors into (18) to find the display-frame target line-of-sight error $\tilde{v}^d$:

$$\tilde{\boldsymbol{v}}^d = \left(-\boldsymbol{\beta} + \boldsymbol{\phi^d} + \boldsymbol{\psi^d}\right) \times \boldsymbol{v}^d. \tag{24}$$

Note from the cross product that the error is orthogonal to the line-of-sight vector $\boldsymbol{v^d} = [1 \quad 0 \quad 0]^T$. Therefore, the line-of-sight angular error in milliradians to the target is simply

$$LOS\ error = \sqrt{\tilde{v}^{d^2}_{\ y} + \tilde{v}^{d^2}_{\ z}} \times 1{,}000\ mrad. \tag{25}$$

5.4 Covariance Analysis of the Line-of-Sight Error

The above provides a means to calculate the boresight error that will result from a particular combination of tracker harmonization and BRU alignment errors. This is useful to understand possible causes for a particular pattern of errors. However, we still need a way to analyse the overall registration accuracy of the Scorpion system in a statistical sense. This can be done through Monte Carlo simulation, or, assuming the errors are small and Gaussian, through covariance analysis.

In this section we develop a covariance analysis for the errors of the line-of-sight vector defined as the x-direction vector of the display-frame. Error in the y component appears as azimuth error, while the z component looks like negative elevation error.

The tracker error over time can be modelled as the sum of a deterministic function of head pose plus a randomly time-varying stationary stochastic process:

$$\boldsymbol{\phi}(\mathrm{t}) = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{w_\varphi}(t). \tag{26}$$

Here $\boldsymbol{f}(\boldsymbol{\theta})$ represents calibration residual tracking errors which are repeatable at different times, but vary as a function of the head euler angles θ. $\boldsymbol{w_\varphi}(t)$ represents a random component of the rotation error vector that wanders around mean zero with a stationary covariance $P_{\varphi\varphi} = \begin{bmatrix}\sigma_\varphi^2 & 0 & 0\\ 0 & \sigma_\varphi^2 & 0\\ 0 & 0 & \sigma_\varphi^2\end{bmatrix}$ and a correlation time of a few tens of seconds so that samples taken minutes apart can be considered independent.

The harmonization error is a random constant that gets determined once during the auto-harmonization procedure and then locked down. Because the p-frame and n-frame are nearly coincident and the pilot is likely to have his head on average pointing straight forward ($\boldsymbol{\theta} = \mathbf{0}$) over the course of the auto-harmonization process, the auto-harmonization algorithm will on average attempt to cancel out the effect of the systematic head-tracker calibration error in the head forward position:

$$\Psi = -\boldsymbol{f}(\boldsymbol{\theta} = \mathbf{0}) + \boldsymbol{w_\psi}, \tag{27}$$

where $\boldsymbol{w_\psi}$ is a random constant representing the variability in the outcome of auto-harmonization with covariance $P_{\psi\psi} = \begin{bmatrix}\sigma_\psi^2 & 0 & 0\\ 0 & \sigma_\psi^2 & 0\\ 0 & 0 & \sigma_\psi^2\end{bmatrix}$.

For simplicity, let us consider the case where boresighting is done to a watermark at zero azimuth and elevation, so that at the time of boresighting, t_{bs}, we again have $\boldsymbol{\theta} = \mathbf{0}$. In that case $C_p^d = I$, so

$$\begin{aligned}\boldsymbol{\beta} &= C_p^d(\boldsymbol{\phi}(t_{bs}) + \boldsymbol{\Psi} - \boldsymbol{\gamma})\\ &= \boldsymbol{\phi}(t_{bs}) + \Psi - \boldsymbol{\gamma}\\ &= \boldsymbol{f}(\boldsymbol{\theta} = \mathbf{0}) + \boldsymbol{w_\varphi}(t_{bs}) - \boldsymbol{f}(\boldsymbol{\theta} = \mathbf{0}) + \boldsymbol{w_\psi} - \boldsymbol{\gamma}\\ &= \boldsymbol{w_\varphi}(t_{bs}) + \boldsymbol{w_\psi} - \boldsymbol{\gamma}.\end{aligned}$$

From (24), the line of sight error at any future time is

$$\tilde{\boldsymbol{v}}^d = [\boldsymbol{q}\times]\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}, \tag{28}$$

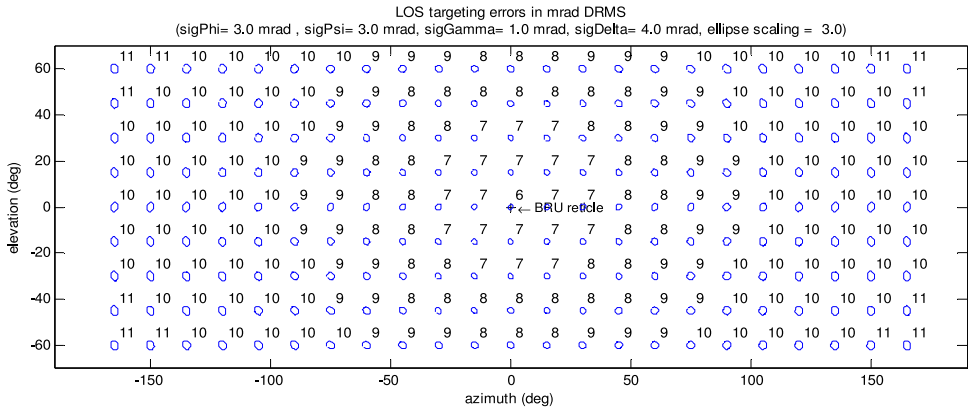

Fig. 17. Symbol registration errors at various azimuth and elevation look directions relative to the vehicle. In each direction, the covariance ellipse for azimuth and elevation error relative to the look direction is shown (scaled up three fold) as well as the two-dimensional RMS error in milliradians.

where

$$\begin{aligned}\boldsymbol{q} &= -\boldsymbol{\beta} + C_p^d(\boldsymbol{\phi}(t) + \Psi) \\ &= -\boldsymbol{w}_{\boldsymbol{\varphi}}(t_{bs}) - \boldsymbol{w}_{\boldsymbol{\psi}} + \boldsymbol{\gamma} + C_p^d\big(\boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{w}_{\boldsymbol{\varphi}}(t) - \boldsymbol{f}(\boldsymbol{\theta}=\mathbf{0}) + \boldsymbol{w}_{\boldsymbol{\psi}}\big) \\ &= \boldsymbol{\gamma} - \boldsymbol{w}_{\boldsymbol{\varphi}}(t_{bs}) + C_p^d\boldsymbol{w}_{\boldsymbol{\varphi}}(t) + (C_p^d - \mathbf{I})\boldsymbol{w}_{\boldsymbol{\psi}} + C_p^d(\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{f}(\boldsymbol{\theta}=\mathbf{0})).\end{aligned} \tag{29}$$

By taking advantage of the statistical independence of each of these terms, we can compute the covariance of $\mathbf{q}$ as

$$\begin{aligned}P_{qq} &= E\big[\boldsymbol{q}\boldsymbol{q}^T\big] \\ &= E\big[\boldsymbol{\gamma}\boldsymbol{\gamma}^T\big] + E\Big[\boldsymbol{w}_{\boldsymbol{\varphi}}\boldsymbol{w}_{\boldsymbol{\varphi}}^T\Big] + C_p^d E\Big[\boldsymbol{w}_{\boldsymbol{\varphi}}\boldsymbol{w}_{\boldsymbol{\varphi}}^T\Big]C_d^p \\ &\quad + \Big(C_p^d - \mathbf{I}\Big)E\Big[\boldsymbol{w}_{\boldsymbol{\psi}}\boldsymbol{w}_{\boldsymbol{\psi}}^T\Big]\big(C_d^p - \mathbf{I}\big) + C_p^d E\big[\delta\mathbf{f}\delta\mathbf{f}^T\big]C_d^p \\ &= P_{\gamma\gamma} + P_{\varphi\varphi} + C_p^d P_{\varphi\varphi}C_d^p + C_p^d P_{\psi\psi}C_d^p - P_{\psi\psi}C_d^p - C_p^d P_{\psi\psi} \\ &\quad + P_{\psi\psi} + C_p^d P_{\delta\delta}(\theta)C_d^p \\ &= P_{\gamma\gamma} + 2\cdot P_{\varphi\varphi} + 2\cdot P_{\psi\psi}\cdot\left(I - \frac{C_d^p + C_p^d}{2}\right) + P_{\delta\delta}(\theta).\end{aligned} \tag{30}$$

The last line is based on an assumption that each covariance matrix is a scalar multiple of the identity matrix and so can commute with the neighbouring rotation matrices. The covariance $P_{\boldsymbol{\delta\delta}}(\boldsymbol{\theta})$ is associated with the residual calibration error $\delta\mathbf{f} = \boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{f}(\boldsymbol{\theta}=\mathbf{0})$ which is zero at the boresight direction and grows with increasing magnitude of the head orientation euler angles vector θ. The function may be a complex function of θ that depends on the details of how the calibration errors vary with head pose and may not grow monotonically, but for simplicity we shall crudely approximate it as linearly dependent on the magnitude of the head rotation away from boresight up to 90 degrees:

$$P_{\delta\delta}(\theta) \approx I\cdot\left(\sigma_\delta\cdot\min\left(\|\theta\|,\frac{\pi}{2}\right)\right)^2. \tag{31}$$

Now that we have the covariance P_{qq}, we can substitute components of that matrix into the covariance matrix for the line-of-sight error, which we obtain from (28) and (29) as follows:

$$\begin{aligned}P_{\mathbf{vv}} &= E\Big[\tilde{\boldsymbol{v}}^d\tilde{\boldsymbol{v}}^{dT}\Big] \\ &= E\left[[\boldsymbol{q}\times]\begin{bmatrix}1\\0\\0\end{bmatrix}[1\quad 0\quad 0][\boldsymbol{q}\times]^T\right] \\ &= E\left[[\boldsymbol{q}\times]\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}[\boldsymbol{q}\times]^T\right] \\ &= E\begin{bmatrix}0&0&0\\0&q_3^2&-q_3q_2\\0&-q_3q_2&q_2^2\end{bmatrix} \\ &= \begin{bmatrix}0&0&0\\0&P_{qq}(3,3)&-P_{qq}(2,3)\\0&-P_{qq}(2,3)&P_{qq}(2,2)\end{bmatrix}.\end{aligned} \tag{32}$$

The bottom right 2X2 submatrix of $P_{\mathbf{vv}}$ is the covariance for the 2d (az,-el) pointing error in the direction of sight. We can pull the elements of this matrix right out of P_{qq}, and we have already derived an expression for that in (30).

A Matlab script is used to plot these 2D LOS error covariance ellipses at a variety of azimuth/elevation look directions. The major and minor axes of the ellipse are determined by finding the eigenvectors of this 2×2 submatrix, and the corresponding eigenvalues are the sigmas along these two directions. The distance-RMS for a multidimensional Gaussian error is defined in [2] as the RSS of the sigmas: $DRMS = \sqrt{\sigma_1^2 + \sigma_2^2}$.

The percentage of the probability enclosed within a circle of radius DRMS varies from 63 percent for a circular distribution ($\sigma_1 = \sigma_2$) to 68 percent for a very elongated distribution ($\sigma_1 > 10\cdot\sigma_2$).

Fig. 17 above shows one run of the Matlab script. The script requires four input parameters that are shown in Table 3. These parameters are used to define the standard deviations of the four contributing covariance matrices in (30).

The values used to generate Fig. 17 are not necessarily representative of the performance of the HObIT and Scorpion in existing aircraft installations, but are intended just for illustrative purposes. A relatively large value of σ_δ shows how dramatically that causes the registration away

TABLE 3
Matlab Script Input Parameters

Parameter	Meaning	Value used in Fig. 17
σ_γ	RMS uncertainty of the alignment of the HUD or BRU which is being used as a reference direction for boresight calibration (HMD-to-tracker)	1 mRad
σ_ϕ	RMS uncertainty of the tracker output rotation matrix C_b^m due to internal calibration errors, noise, etc.	3 mRad
σ_ψ	RMS uncertainty of the harmonization alignment angles C_n^p set during the auto-harmonization, possibly on a previous flight.	3 mRad
σ_δ	RMS uncertainty of the residual calibration error at 90° off boresight, applied linearly as in (31)	4 mRad

from the boresight direction to degrade. Conversely, 1 mRad for HUD alignment is probably a bit optimistic, although increasing that slightly would not make much difference to the RSS if it is still one of the smaller terms. The error ellipses in the figure are scaled up by a factor of 3 to make them more visible, with their DRMS values printed above each ellipse.

6 Discussion

The work presented in this paper has resulted in a novel AR system for use in moving vehicles. The system meets all its performance targets and has been verified through thorough simulations, car testing, and is now undergoing flight test programs on several platforms. In the coming year the software will be developed to meet the Federal Aviation Administration's DO-178C software standards for a qualified airworthy system.

There are also potential AR applications in ground vehicles, as perhaps suggested by the demo video filmed in a car. One interesting such application occurs in climates where the snow piles so high that ploughs cannot see the tops of the roadside guidance poles. It seems inevitable that AR guided plows will eventually be deployed. Another very alluring possibility is AR guidance for operators of earth-moving equipment and other heavy construction machinery to help them more efficiently and exactly achieve the desired results planned in a CAD model. Obviously AR headsets would also be valuable to provide situational awareness to first responders while they drive to the scene, and after they dismount from the vehicles at the scene.

The novel auto-harmonization algorithm presented makes it possible to consider deploying the HObIT on these types of vehicles, which are not equipped with a HUD to facilitate traditional methods of harmonization. Of course many of these vehicles do not currently have installed GPS/INS systems, but with the recent developments in the MEMS field, the cost of sufficient performance is falling very rapidly and it may already be practical to add a MEMS GPS/INS unit as part of the vehicular AR package.

The covariance-based analysis of line-of-sight registration error in Section 5 provides a new approach to error budgeting for AR systems that may involve user calibration procedures that tend to cancel out errors in a particular direction. It is a bit complicated by the extra coordinate frames in the vehicular application, but we hope it will inspire others to develop a simplified version for more general AR systems on the ground.

References

[1] R. Atac and E. Foxlin, "Scorpion hybrid optical-based inertial tracker (HObIT)," *Proc. SPIE, Head- and Helmet-Mounted Displays XVIII: Design and Applications*, vol. 8735, p. 873502, May 16, 2013.

[2] P. Axelrad, "Notes on positioning errors," *From Course Notes of ASEN5190 GPS Technology, University of Colorado, Fall* 2001, [Online]. Available: http://www.colorado.edu/ASEN/asen5190/error_notes.pdf

[3] R. Azuma, and G. Bishop, "Improving static and dynamic registration in a see-through HMD," in *Proc. 21st Annu. Conf. Comput. Graph. Interactive Techn.*, Orlando, FL, USA, 1994, pp. 197–204.

[4] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed. New York, NY, USA: Wiley, 1992.

[5] K. Dorfmuller, "Robust tracking for augmented reality using retroreflective markers," *Comput. Graph.*, vol. 23, pp. 795–800, 1999.

[6] F. Ferrin, "Survey of helmet tracking technologies," in *Proc. SPIE*, vol. 1456, pp. 86–94, 1991.

[7] E. Foxlin and L. Naimark, "VIS-Tracker: A wearable vision-inertial self-tracker," in *Proc. IEEE Conf. Virtual Reality*, Los Angeles, CA, USA, 2003, pp. 199–206.

[8] E. Foxlin, Y. Altshuler, L Naimark, and M. Harrington, "FlightTracker: A novel optical/inertial tracker for cockpit enhanced vision," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Reality*, Nov. 2–5, 2004, pp. 212–221.

[9] E. Foxlin, "Head-tracking relative to a moving vehicle or simulator platform using differential inertial sensors," presented at the Helmet Head-Mounted Displays V, SPIE, AeroSense Symp., Orlando, FL, USA, Apr. 24–25, 2000.

[10] E. Foxlin, T. Calloway, and H. Zhang, "Improved registration for vehicular AR using auto-harmonization," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Reality*, Munich, Germany, Sep. 2014, pp. 105–112.

[11] T. A. Furness, "The super cockpit and human factors challenges," in *Proc. Human Factors Soc. 30th Annu. Meeting*, 1986, pp. 48–52.

[12] D. W. F. Krevelen, and R. Poelman, "A survey of augmented reality technologies," *Appl. Limitations. Int. J. Virtual Reality*, vol. 9, no. 2, pp. 1–20, 2010.

[13] G. Moussa, E. Radwan, and K. Hussain, "Augmented reality vehicle system: Left-turn maneuver study," *Transp. Res. Part C: Emerging Technol.*, vol. 21, no. 1, pp. 1–16, 2012.

[14] D. Nister, "An efficient solution to the five-point relative pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[15] D. Roberts, A. Menozzi, J. Cook, T. Sherrill, S. Snarski, P. Russler, and G. Welch, "Testing and evaluation of a wearable augmented reality system for natural outdoor environments," in *Proc. SPIE Defense, Security, Sensing, Int. Soc. Opt. Photon.*, vol. 8735, p. 87350A, May 2013.

[16] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Stevenage, U.K.: IET, Jan. 2004.

[17] M. Tonnis, C. Sandor, C. Lange, and H. Bubb, "Experimental evaluation of an augmented reality visualization for directing a car driver's attention," *Proc. 4th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2005, pp. 56–59.

[18] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2008, pp. 125–134.

[19] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller and D. Colluci, "High-performance wide-area optical tracking: The hiball tracking system," *Presence: Teleoperators Virtual Environ.*, vol. 10, no. 1, pp. 1–21, 2001.

[20] R. Wood and P. Howells, "Chapter 4: Head-up displays," in *The Avionics Handbook*, C. Spitzer, ed. Boca Raton, FL, USA: CRC Press, 2001.

[21] F. Zhou, D. H. Been-Lirn, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2008, pp. 193–202.
[22] B. Jau, "Technical support Package on anthropomorphic remote manipulator for NASA TECH BRIEF," JPL Technol. Utilization Office, Pasadena, CA, USA, 15(4) from JPL Invention Rep. (Rep. No. NPO- 17975/7222), 1991.
[23] Y. J. Lee, C. G. Park, and S. K. Hong, "Helmet tracker system using stereo cameras," in *Proc. Int. Joint Conf.*, Oct. 2006, pp. 3636–3640.

Eric Foxlin received the BS degree in physics from Harvard in 1987, and the MS degree in electrical engineering and computer science from MIT in 1993. He founded InterSense (now part of Thales Visionix) in 1996, based on inertial tracking technology he developed at MIT. He is currently a lead research scientist at Fitbit in Boston, where his focus is on wearable technology, signal processing, and machine learning to improve health.

Thomas Calloway received the BS degree in electrical engineering in 2005, and the MS degree in computer engineering in 2011, both from the University of Massachusetts in Lowell. He has more than 10 years of motion tracking experience and is currently employed as the director in Advanced Programs at Thales Visionix in Billerica Massachusetts.

Hongsheng Zhang received the BE degree from Xi'an Jiaotong University in 1996, the ME degree from Tsinghua University in 1999, and the PhD degree from the University of Miami in 2005. He is currently a computer vision researcher with Thales Visionix and lives in Wellesley MA.